

## Final Year Project

- Topic : Text-to-Image Generation with Deep Learning Models: Exploring the Potential of Multimodal and Generative Model

Submitted By : Subrat Kumar Nanda

Roll No : 22DS011003

Under the guidance of : Prof. Dr. Suman Pathak

```
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive + Code + Text

#!pip install --upgrade nvidia-pyindex

Requirement already satisfied: nvidia-pyindex in /usr/local/lib/python3.10/dist-packages (1.0.9)

#!pip install mxnet-cu80

Requirement already satisfied: mxnet-cu80 in /usr/local/lib/python3.10/dist-packages (1.5.0)
Requirement already satisfied: numpy<2.0.0,>1.16.0 in /usr/local/lib/python3.10/dist-packages (from mxnet-cu80) (1.25.2)
Requirement already satisfied: requests<3,>=2.20.0 in /usr/local/lib/python3.10/dist-packages (from mxnet-cu80) (2.31.0)
Requirement already satisfied: graphviz<0.9.0,>=0.8.1 in /usr/local/lib/python3.10/dist-packages (from mxnet-cu80) (0.8.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.20.0->mxnet-cu80) (3.7)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.20.0->mxnet-cu80) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.20.0->mxnet-cu80) (2.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.20.0->mxnet-cu80) (202

!nvcc --version

nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2023 NVIDIA Corporation
Built on Tue_Aug_15_22:02:13_PDT_2023
Cuda compilation tools, release 12.2, V12.2.140
Build cuda_12.2.r12.2/compiler.33191640_0

!nvidia-smi 2>&1 | grep -i "cuda"
!nvidia-smi
# Give you information about graphic card & this nvidia library for gpu
# Driver is cuda version:12.0
# How much memory gpu has 15360mb which is 15 GB & i am using 0 GB

| NVIDIA-SMI 535.104.05      Driver Version: 535.104.05    CUDA Version: 12.2      |
Tue Jul  2 19:50:22 2024
+-----+
| NVIDIA-SMI 535.104.05      Driver Version: 535.104.05    CUDA Version: 12.2      |
+-----+
| GPU  Name        Persistence-M | Bus-Id     Disp.A  | Volatile Uncorr. ECC  | |
| Fan  Temp  Perf  Pwr:Usage/Cap | Memory-Usage | GPU-Util  Compute M.  |
|                               |             |            | MIG M.               |
+-----+
|   0  Tesla T4           Off  | 00000000:00:04.0 Off |          0 | |
| N/A   35C    P8          10W /  70W |      0MiB / 15360MiB |     0%      Default |
|                               |             |            | N/A                 |
+-----+
+-----+
| Processes:                   GPU Memory  |
| GPU  GI  CI      PID  Type  Process name        Usage  |
| ID  ID              |          |             |          |
+-----+
| No running processes found  |
+-----+
```

```
# lets create function

def process_user(name):
    when = 'today'
    print(name, 'is using NoteBook', when)
process_user('Subrat Nanda')
```

→ Subrat Nanda is using NoteBook today

## ✓ how to debugging above code in gpu

- import pdb
- Python Debugger module.PDB is a built-in interactive debugger for Python programs.
- It allows developers to set breakpoints, step through code, inspect variables, and perform other debugging tasks to help identify and fix issues in their code.
- Refer to chatgpt (l - active line number, n - next line)

```
import pdb

def process_nit(name):
    #pdb.set_trace()
    when = 'today'
    print(name, 'is using NoteBook', when)

process_nit('Subrat')

→ Subrat is using NoteBook today
```

- LETS WORK ON MULTIMODEL DEEP LEARNING ARCHITECTURE CALLED CLIP WHICH LINK TEXT WITH VISUAL ELIMINATION WE ARE COMBINE WITH GENERATIVE MODEL WITH TRANSFORMER TYPE OF ARCHITECTURE.
- WE WILL ABLE TO TAKE TEXT PROMPT AND GENERATE VISUALIZE IMAGE AND VIDEO AND VIDEO SEQUENCE FROM THE TEXT PROMPT.
- THIS IS THE CUTTING EDGE.

```
# CLIP ARCHITECTURE
!git clone https://github.com/openai/CLIP.git
%cd /content/CLIP/

→ Cloning into 'CLIP'...
remote: Enumerating objects: 256, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 256 (delta 6), reused 4 (delta 0), pack-reused 243
Receiving objects: 100% (256/256), 8.93 MiB | 16.93 MiB/s, done.
Resolving deltas: 100% (130/130), done.
/content/CLIP
```

# TAMING-TRANSFORMER ARCHITECTURE

```
!git clone https://github.com/CompVis/taming-transformers

→ Cloning into 'taming-transformers'...
remote: Enumerating objects: 1342, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 1342 (delta 0), reused 1 (delta 0), pack-reused 1340
Receiving objects: 100% (1342/1342), 409.77 MiB | 28.41 MiB/s, done.
Resolving deltas: 100% (282/282), done.
```

# We Need to install some more libraries as well

```
!pip install --no-deps ftfy regex tqdm
!pip install omegaconf==2.0.0 pytorch-lightning==1.0.8
!pip uninstall torchtext --yes
!pip install einops
```

→ Collecting ftfy  
  Downloading ftfy-6.2.0-py3-none-any.whl (54 kB)

54.4/54.4 kB 2.9 MB/s eta 0:00:00

Requirement already satisfied: regex in /usr/local/lib/python3.10/dist-packages (2024.5.15)  
 Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (4.66.4)  
 Installing collected packages: ftfy  
 Successfully installed ftfy-6.2.0  
 Collecting omegaconf==2.0.0  
 Downloading omegaconf-2.0.0-py3-none-any.whl (33 kB)  
 Collecting pytorch-lightning==1.0.8  
 Downloading pytorch\_lightning-1.0.8-py3-none-any.whl (561 kB)

561.4/561.4 kB 9.6 MB/s eta 0:00:00

Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from omegaconf==2.0.0) (6.0.1)  
 Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from omegaconf==2.0.0) (4.12.2)  
 Requirement already satisfied: numpy>=1.16.4 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (1.25.2)  
 Requirement already satisfied: torch>=1.3 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (2.3.0+cu121)  
 Requirement already satisfied: future>=0.17.1 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (0.18.3)  
 Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (4.66.4)  
 Requirement already satisfied: fsspec>=0.8.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (2023.6.0)  
 Requirement already satisfied: tensorboard>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning==1.0.8) (2.15.2)  
 Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: setupools>=41.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: six>1.9 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning==1)  
 Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (3.15.1)  
 Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (1.12.1)  
 Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (3.3)  
 Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.3->pytorch-lightning==1.0.8) (3.1.4)  
 Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_cuda\_nvrtc\_cu12-12.1.105-py3-none-manylinux1\_x86\_64.whl (23.7 MB)  
 Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_cuda\_runtime\_cu12-12.1.105-py3-none-manylinux1\_x86\_64.whl (823 kB)  
 Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_cuda\_cupti\_cu12-12.1.105-py3-none-manylinux1\_x86\_64.whl (14.1 MB)  
 Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_cudnn\_cu12-8.9.2.26-py3-none-manylinux1\_x86\_64.whl (731.7 MB)  
 Collecting nvidia-cublas-cu12==12.1.3.1 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_cublas\_cu12-12.1.3.1-py3-none-manylinux1\_x86\_64.whl (410.6 MB)  
 Collecting nvidia-cufft-cu12==11.0.2.54 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_cufft\_cu12-11.0.2.54-py3-none-manylinux1\_x86\_64.whl (121.6 MB)  
 Collecting nvidia-curand-cu12==10.3.2.106 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_curand\_cu12-10.3.2.106-py3-none-manylinux1\_x86\_64.whl (56.5 MB)  
 Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_cusolver\_cu12-11.4.5.107-py3-none-manylinux1\_x86\_64.whl (124.2 MB)  
 Collecting nvidia-cusparse-cu12==12.1.0.106 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_cusparse\_cu12-12.1.0.106-py3-none-manylinux1\_x86\_64.whl (196.0 MB)  
 Collecting nvidia-nccl-cu12==2.20.5 (from torch>=1.3->pytorch-lightning==1.0.8)  
 Using cached nvidia\_nccl\_cu12-2.20.5-py3-none-manylinux2014\_x86\_64.whl (176.2 MB)  
 Collecting nvidia-nvtx-cu12==12.1.105 (from torch>=1.3->pytorch-lightning==1.0.8)

# import IMAGE, NUMPY,PANDAS,MATPLTOLIB libraries

```
import PIL
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
# import PYTORCH libraries
import torch, os, imageio, pdb, math
import torchvision
import torchvision.transforms as T
import torchvision.transforms.functional as TF
```

```
import yaml
from omegaconf import OmegaConf

from CLIP import clip

import warnings
warnings.filterwarnings('ignore')
```

```

## helper functions

def show_from_tensor(tensor):
    img = tensor.clone()
    img = img.mul(255).byte()
    img = img.cpu().numpy().transpose((1,2,0))

    plt.figure(figsize=(10,7))
    plt.axis('off')
    plt.imshow(img)
    plt.show()

def norm_data(data):
    return (data.clip(-1,1)+1)/2 ### range between 0 and 1 in the result

### Parameters
learning_rate = .5
batch_size = 1
wd = .1
noise_factor = .22
init_image = None

total_iter=400
im_shape = [450, 450, 3] # height, width, channel
size1, size2, channels = im_shape

### CLIP MODEL ###
clipmodel, _ = clip.load('ViT-B/32', jit=False)
clipmodel.eval()
print(clip.available_models())

print("Clip model visual input resolution: ", clipmodel.visual.input_resolution)

device=torch.device("cuda:0")
torch.cuda.empty_cache()

→ 100%|██████████| 338M/338M [00:07<00:00, 45.3MiB/s]
['RN50', 'RN101', 'RN50x4', 'RN50x16', 'RN50x64', 'ViT-B/32', 'ViT-B/16', 'ViT-L/14', 'ViT-L/14@336px']
Clip model visual input resolution: 224

!pip install taming-transformers-rom1504

→ Collecting taming-transformers-rom1504
  Downloading taming_transformers_rom1504-0.0.6-py3-none-any.whl (51 kB) 51.5/51.5 kB 2.6 MB/s eta 0:00:00
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (from taming-transformers-rom1504) (2.3.0+cu121)
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (from taming-transformers-rom1504) (0.18.0+cu121)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from taming-transformers-rom1504) (1.25.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from taming-transformers-rom1504) (4.66.4)
Requirement already satisfied: omegaconf>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from taming-transformers-rom1504) (2.0.0)
Requirement already satisfied: pytorch-lightning>=1.0.8 in /usr/local/lib/python3.10/dist-packages (from taming-transformers-rom1504)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from omegaconf>=2.0.0->taming-transformers-rom1504)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from omegaconf>=2.0.0->taming-transformers-rom1504)
Requirement already satisfied: future>=0.17.1 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning>=1.0.8->taming-transformers-rom1504)
Requirement already satisfied: fsspec>=0.8.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning>=1.0.8->taming-transformers-rom1504)
Requirement already satisfied: tensorboard>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning>=1.0.8->taming-transformers-rom1504)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504) (3.15.4)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504) (1.12.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504) (3.1.4)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-nccl-cu12==2.20.5 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504)
Requirement already satisfied: triton==2.3.0 in /usr/local/lib/python3.10/dist-packages (from torch->taming-transformers-rom1504) (2.3.0)
Requirement already satisfied: nvidia-nvjitlink-cu12 in /usr/local/lib/python3.10/dist-packages (from nvidia-cusolver-cu12==11.4.5.107)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.10/dist-packages (from torchvision->taming-transformers-rom1504)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.0.0->pytorch-lightning>=1.12.1)
Requirement already satisfied: grpcio>=1.48.2 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.0.0->pytorch-lightning>=1.12.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.0.0->pytorch-lightning>=1.12.1)
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.0.0->pytorch-lightning>=1.12.1)

```

```
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning)
Requirement already satisfied: protobuf!=4.24.0,>=3.19.6 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch)
Requirement already satisfied: requests<3,>>2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-light
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightn
Requirement already satisfied: six>1.9 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning)>=1.0.8-
Requirement already satisfied: tensorboard-data-server<0.8.0,>>0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard>=2.2.0->pytorch-lightning
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch->taming-transformers-ro
Requirement already satisfied: mpmath<1.4.0,>>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch->taming-transformer
Requirement already satisfied: cachetools<6.0,>>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>>=1.6.3->tensorb
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>>=1.6.3->tensorbo
Requirement already satisfied: rsa<5,>>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>>=1.6.3->tensorboard>=2.2
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<2,>>=0.5
Requirement already satisfied: charset-normalizer<4,>>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>>=2.21.0->tensorb
Requirement already satisfied: idna<4,>>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>>=2.21.0->tensorboard>=2.2.0-
Requirement already satisfied: urllib3<3,>>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>>=2.21.0->tensorboard>=
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>>=2.21.0->tensorboard>=
Requirement already satisfied: pyasn1<0.7.0,>>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-au
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-auth
Installing collected packages: taming-transformers-rom1504
-----
```

```
## Taming transformer instantiation
```

```
%cd taming-transformers/
```

```
!mkdir -p models/vqgan_imagenet_f16_16384/checkpoints
!mkdir -p models/vqgan_imagenet_f16_16384/configs

if len(os.listdir('models/vqgan_imagenet_f16_16384/checkpoints')) == 0:
    !wget 'https://heibox.uni-heidelberg.de/f/867b05fc8c4841768640/?dl=1' -O 'models/vqgan_imagenet_f16_16384/checkpoints/last.ckpt'
    !wget 'https://heibox.uni-heidelberg.de/f/274fb24ed38341bfa753/?dl=1' -O 'models/vqgan_imagenet_f16_16384/configs/model.yaml'
```

```
→ /content/CLIP/taming-transformers
--2024-07-02 19:53:09-- https://heibox.uni-heidelberg.de/f/867b05fc8c4841768640/?dl=1
Resolving heibox.uni-heidelberg.de (heibox.uni-heidelberg.de)... 129.206.7.113
Connecting to heibox.uni-heidelberg.de (heibox.uni-heidelberg.de)|129.206.7.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://heibox.uni-heidelberg.de/seafhttp/files/41627cc7-b717-429a-9db0-2827b20c376e/last.ckpt [following]
--2024-07-02 19:53:09-- https://heibox.uni-heidelberg.de/seafhttp/files/41627cc7-b717-429a-9db0-2827b20c376e/last.ckpt
Reusing existing connection to heibox.uni-heidelberg.de:443.
HTTP request sent, awaiting response... 200 OK
Length: 980092370 (935M) [application/octet-stream]
Saving to: 'models/vqgan_imagenet_f16_16384/checkpoints/last.ckpt'

models/vqgan_imagen 100%[=====] 934.69M 11.3MB/s in 89s
```

```
2024-07-02 19:54:39 (10.5 MB/s) - 'models/vqgan_imagenet_f16_16384/checkpoints/last.ckpt' saved [980092370/980092370]
```

```
--2024-07-02 19:54:39-- https://heibox.uni-heidelberg.de/f/274fb24ed38341bfa753/?dl=1
Resolving heibox.uni-heidelberg.de (heibox.uni-heidelberg.de)... 129.206.7.113
Connecting to heibox.uni-heidelberg.de (heibox.uni-heidelberg.de)|129.206.7.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://heibox.uni-heidelberg.de/seafhttp/files/1e5e6b53-c769-4b7f-88f7-5c1edc7201de/model.yaml [following]
--2024-07-02 19:54:39-- https://heibox.uni-heidelberg.de/seafhttp/files/1e5e6b53-c769-4b7f-88f7-5c1edc7201de/model.yaml
Reusing existing connection to heibox.uni-heidelberg.de:443.
HTTP request sent, awaiting response... 200 OK
Length: 692 [application/octet-stream]
Saving to: 'models/vqgan_imagenet_f16_16384/configs/model.yaml'
```

```
models/vqgan_imagen 100%[=====] 692 --.-KB/s in 0s
```

```
2024-07-02 19:54:39 (648 MB/s) - 'models/vqgan_imagenet_f16_16384/configs/model.yaml' saved [692/692]
```

```

from taming.models.vqgan import VQModel

def load_config(config_path, display=False):
    config_data = OmegaConf.load(config_path)
    if display:
        print(yaml.dump(OmegaConf.to_container(config_data)))
    return config_data

def load_vqgan(config, chk_path=None):
    model = VQModel(**config.model.params)
    if chk_path is not None:
        state_dict = torch.load(chk_path, map_location="cpu")["state_dict"]
        missing, unexpected = model.load_state_dict(state_dict, strict=False)
    return model.eval()

def generator(x):
    x = taming_model.post_quant_conv(x)
    x = taming_model.decoder(x)
    return x

...
# Check if CUDA is available and set device accordingly
if torch.cuda.is_available():
    device = torch.device("cuda")
else:
    device = torch.device("cpu")
    print("Warning: CUDA not available, using CPU instead.")
...
taming_config = load_config("./models/vqgan_imagenet_f16_16384/configs/model.yaml", display=True)
taming_model = load_vqgan(taming_config, chk_path="./models/vqgan_imagenet_f16_16384/checkpoints/last.ckpt").to(device)

```

```

→ model:
  base_learning_rate: 4.5e-06
  params:
    ddconfig:
      attn_resolutions:
        - 16
      ch: 128
      ch_mult:
        - 1
        - 1
        - 2
        - 2
        - 4
      double_z: false
      dropout: 0.0
      in_channels: 3
      num_res_blocks: 2
      out_ch: 3
      resolution: 256
      z_channels: 256
    embed_dim: 256
    lossconfig:
      params:
        codebook_weight: 1.0
        disc_conditional: false
        disc_in_channels: 3
        disc_num_layers: 2
        disc_start: 0
        disc_weight: 0.75
        target: taming.modules.losses.vqperceptual.VQLPIPSWithDiscriminator
      monitor: val/rec_loss
      n_embed: 16384
    target: taming.models.vqgan.VQModel

```

Working with z of shape (1, 256, 16, 16) = 65536 dimensions.

Downloading: "<https://download.pytorch.org/models/vgg16-397923af.pth>" to /root/.cache/torch/hub/checkpoints/vgg16-397923af.pth  
100% [██████████] 528M/528M [00:05<00:00, 101MB/s]

Downloading vgg\_lpips model from <https://heibox.uni-heidelberg.de/f/607503859c864bc1b30b/?dl=1> to taming/modules/autoencoder/lpip...  
8.19KB [00:00, 5.95MB/s] loaded pretrained LPIPS loss from taming/modules/autoencoder/lpip.../vgg.pth

VQLPIPSWithDiscriminator running with hinge loss.

```

### Declare the values that we are going to optimize

class Parameters(torch.nn.Module):
    def __init__(self):
        super(Parameters, self).__init__()
        self.data = .5*torch.randn(batch_size, 256, size1//16, size2//16).cuda() # 1x256x14x15 (225/16, 400/16)
        self.data = torch.nn.Parameter(torch.sin(self.data))

    def forward(self):
        return self.data

def init_params():
    params=Parameters().cuda()
    optimizer = torch.optim.AdamW([{'params': [params.data]}, {'lr': learning_rate}], weight_decay=wd)
    return params, optimizer

### Encoding prompts and a few more things
normalize = torchvision.transforms.Normalize((0.48145466, 0.4578275, 0.40821073), (0.26862954, 0.26130258, 0.27577711))

def encodeText(text):
    t=clip.tokenize(text).cuda()
    t=clipmodel.encode_text(t).detach().clone()
    return t

def createEncodings(include, exclude, extras):
    include_enc=[]
    for text in include:
        include_enc.append(encodeText(text))
    exclude_enc=encodeText(exclude) if exclude != '' else []
    extras_enc=encodeText(extras) if extras !='' else []
    return include_enc, exclude_enc, extras_enc

augTransform = torch.nn.Sequential(
    torchvision.transforms.RandomHorizontalFlip(),
    torchvision.transforms.RandomAffine(30, (.2, .2), fill=0)
).cuda()

Params, optimizer = init_params()

with torch.no_grad():
    print(Params().shape)
    img= norm_data(generator(Params()).cpu()) # 1 x 3 x 224 x 400 [225 x 400]
    print("img dimensions: ",img.shape)
    show_from_tensor(img[0])

```

```
→ torch.Size([1, 256, 28, 28])
img dimensions: torch.Size([1, 3, 448, 448])
```



```
### create crops

def create_crops(img, num_crops=32):
    p=size1//2
    img = torch.nn.functional.pad(img, (p,p,p,p), mode='constant', value=0) # 1 x 3 x 448 x 624 (adding 112*2 on all sides to 224x400)

    img = augTransform(img) #RandomHorizontalFlip and RandomAffine

    crop_set = []
    for ch in range(num_crops):
        gap1= int(torch.normal(1.2, .3, ()).clip(.43, 1.9) * size1)
        offsetx = torch.randint(0, int(size1*2-gap1),())
        offseyt = torch.randint(0, int(size1*2-gap1),())

        crop=img[:, :, offsetx:offsetx+gap1, offseyt:offseyt+gap1]

        crop = torch.nn.functional.interpolate(crop,(224,224), mode='bilinear', align_corners=True)
        crop_set.append(crop)

    img_crops=torch.cat(crop_set,0) ## 30 x 3 x 224 x 224

    randnormal = torch.randn_like(img_crops, requires_grad=False)
    num_rands=0
    randstotal=torch.rand((img_crops.shape[0],1,1,1)).cuda() #32

    for ns in range(num_rands):
        randstotal*=torch.rand((img_crops.shape[0],1,1,1)).cuda()

    img_crops = img_crops + noise_factor*randstotal*randnormal

    return img_crops
```

```
### Show current state of generation
```

```
def showme(Params, show_crop):
    with torch.no_grad():
        generated = generator(Params())

    if (show_crop):
        print("Augmented cropped example")
        aug_gen = generated.float() # 1 x 3 x 224 x 400
        aug_gen = create_crops(aug_gen, num_crops=1)
        aug_gen_norm = norm_data(aug_gen[0])
        show_from_tensor(aug_gen_norm)

    print("Generation")
    latest_gen=norm_data(generated.cpu()) # 1 x 3 x 224 x 400
    show_from_tensor(latest_gen[0])

return (latest_gen[0])
```

## ▼ calculate the Inception Score

```
# Import necessary libraries for Inception and FID calculation
import torch
import torch.nn.functional as F
from torchvision.models import inception_v3
from scipy.stats import entropy
from scipy.linalg import sqrtm
import numpy as np
import torch.nn as nn

# Load pre-trained Inception model for Inception Score
inception_model = inception_v3(pretrained=True, transform_input=False).cuda()
inception_model.eval()
up = nn.Upsample(size=(299, 299), mode='bilinear').cuda()

def inception_score(images, classifier_model):
    """
    Calculate the Inception Score for generated images.

    Args:
    - images: Generated images (30 x 3 x 224 x 224).
    - classifier_model: Inception model for scoring.

    Returns:
    - inception_score: Calculated Inception Score.
    """
    batch_size = images.size(0)
    images_upsampled = up(images)
    logits = classifier_model(images_upsampled)
    probs = F.softmax(logits, dim=1)
    p_yx = probs.mean(dim=0)
    p_y = torch.ones_like(p_yx) / p_yx.size(0)
    kl_div = F.kl_div(probs.log(), p_y, reduction='batchmean')
    inception_score = torch.exp(kl_div)
    return inception_score.item()
```

→ Downloading: "[https://download.pytorch.org/models/inception\\_v3\\_google-0cc3c7bd.pth](https://download.pytorch.org/models/inception_v3_google-0cc3c7bd.pth)" to /root/.cache/torch/hub/checkpoints/inception\_v3\_€  
100%|██████████| 104M/104M [00:00<00:00, 123MB/s]

## ▼ Explanation to below Functions -

`optimizer_result`:

Calculates the optimized loss for CLIP-VQGAN image generation,  
including optional Inception Score estimation.

Args:

Params: A function returning the parameters for the VQGAN generator.  
prompt: The text description for the desired image.

```
clip_model: The pre-trained CLIP model (text and image encoders).
device: The device (CPU or GPU) for computations.
```

Returns:

```
A tuple containing:
- final_loss: The combined loss for optimization.
- inception_score (optional): The estimated Inception Score (if calculated).
```

Optimizers :

```
Optimizes the VQGAN parameters to generate an image matching the prompt.
```

Args:

```
Params: A function returning the parameters for the VQGAN generator.
optimizer: The optimizer (e.g., Adam) for parameter updates.
prompt: The text description for the desired image.
clip_model: The pre-trained CLIP model (text and image encoders).
device: The device (CPU or GPU) for computations.
```

Returns:

```
# Modify optimize_result function to calculate Inception Score and FID Score
from numpy import mean
def optimize_result(Params, prompt):
    alpha = 1 # the importance of the include encodings
    beta = 0.5 # the importance of the exclude encodings
    is_scores = list()
    # Generate images
    out = generator(Params())
    out = norm_data(out)
    out = create_crops(out)
    out = normalize(out) # 30 x 3 x 224 x 224
    image_enc = clipmodel.encode_image(out) # 30 x 512

    # Text encoding
    final_enc = w1 * prompt + w1 * extras_enc # prompt and extras_enc : 1 x 512
    final_text_include_enc = final_enc / final_enc.norm(dim=-1, keepdim=True) # 1 x 512
    final_text_exclude_enc = exclude_enc

    # Calculate cosine similarities
    main_similarity = torch.cosine_similarity(final_text_include_enc, image_enc, dim=-1) # 30
    penalize_similarity = torch.cosine_similarity(final_text_exclude_enc, image_enc, dim=-1) # 30

    # Calculate Inception Score
    with torch.no_grad():
        inception_score_value = inception_score(out, inception_model)
        inception_score_value = inception_score(out.cuda(), inception_model)
        is_scores.append(inception_score_value)

    # Calculate the loss
    main_loss = alpha * main_similarity.mean() # average over the batch of images
    penalize_loss = -beta * penalize_similarity.mean() # average over the batch of images

    final_loss = -alpha*main_loss + beta*penalize_loss

    final_loss = final_loss.mean()
    main_similarity = main_similarity.mean()
    penalize_similarity = penalize_similarity.mean()
    inception_score_value = mean(is_scores)
    #print ("My code : Inception score : {}", inception_score_value)
    return final_loss, main_similarity, penalize_similarity, inception_score_value

# Define the optimize function
def optimize(Params, optimizer, prompt):
    loss, cosine_similarity_score, penalize_similarity, inception_score_value = optimize_result(Params, prompt)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
    return loss, inception_score_value, cosine_similarity_score
```

```
# Modify training_loop to include FID Score calculation
def training_loop(Params, optimizer, show_crop=False):
    res_img = []
    res_z = []

    for prompt in include_enc:
        iteration = 0
        Params, optimizer = init_params() # 1 x 256 x 14 x 25 (225/16, 400/16)

        for it in range(total_iter):
            loss,inception_score_value, cosine_similarity_score = optimize(Params, optimizer, prompt)

            if iteration >= 80 and iteration % show_step == 0:
                new_img = showme(Params, show_crop)
                res_img.append(new_img)
                res_z.append(Params()) # 1 x 256 x 14 x 25
                print("loss:", loss.item(), "\niteration:", iteration)
                print("cosine_similarity:", cosine_similarity_score.item(), "\niteration:", iteration)
                print("Inception Score:", inception_score_value.item(), "\niteration:", iteration)

            iteration += 1
        torch.cuda.empty_cache()
    return res_img, res_z
```

#To Clear CUDA Cache Memory \*Note this is an optional Step.  
`torch.cuda.empty_cache()`

## Generating Image from TEXT input

```
torch.cuda.empty_cache()
include=['A BLUE TREE IN THE FOREST']
exclude='watermark'
extras = ""
w1=1
w2=1
noise_factor=.22
total_iter=110
show_step=10 # set this to see the result every 10 interations beyond iteration 80
include_enc, exclude_enc, extras_enc = createEncodings(include, exclude, extras)
res_img, res_z=training_loop(Params, optimizer, show_crop=True)
```

Augmented cropped example

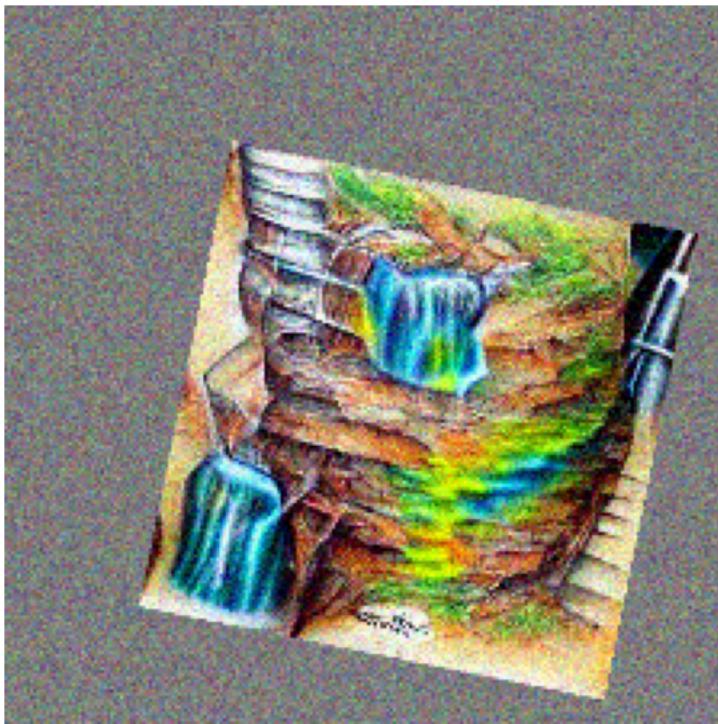


Generation

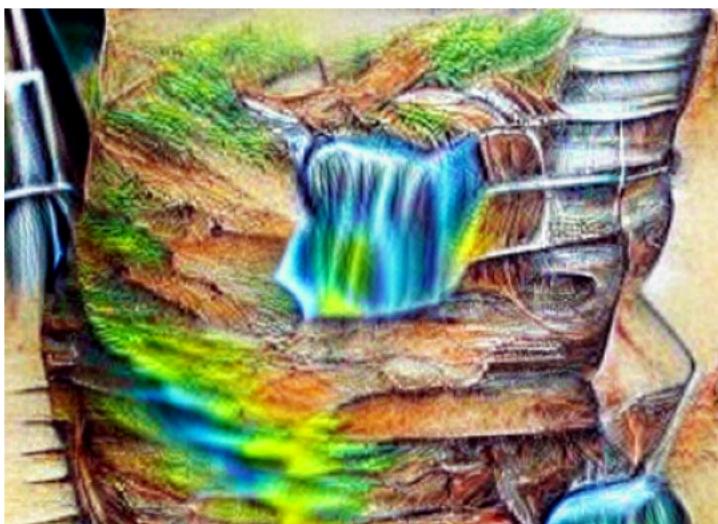


```
torch.cuda.empty_cache()
include=['A colored pencil drawing of a waterfall']
exclude='watermark'
extras = ""
w1=1
w2=1
noise_factor=.20
total_iter=110
show_step=10 # set this to see the result every 10 interations beyond iteration 80
include_enc, exclude_enc, extras_enc = createEncodings(include, exclude, extras)
res_img, res_z=training_loop(Params, optimizer, show_crop=True)
```

Augmented cropped example

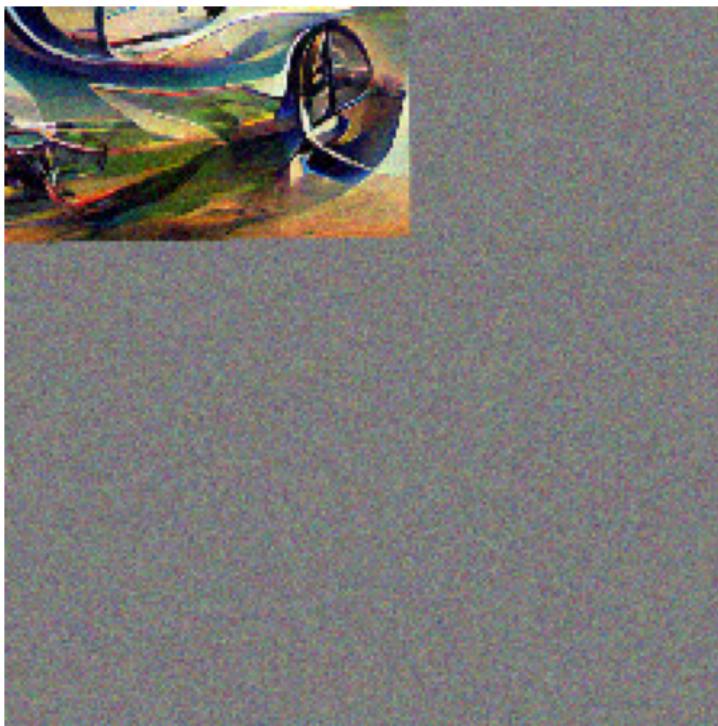


Generation



```
torch.cuda.empty_cache()
include=[' an autogyro flying car, trending on artstation']
exclude='watermark'
extras = ""
w1=1
w2=1
noise_factor=.20
total_iter=110
show_step=10 # set this to see the result every 10 interations beyond iteration 80
include_enc, exclude_enc, extras_enc = createEncodings(include, exclude, extras)
res_img, res_z=training_loop(Params, optimizer, show_crop=True)
```

Augmented cropped example



Generation



```
torch.cuda.empty_cache()
include=['A BLUE TREE IN THE FOREST', 'KIDS PLAYING IN MOON', 'FLOWERS DANCING']
exclude='watermark'
extras = ""
w1=1
w2=1
noise_factor=.22
total_iter=110
show_step=10 # set this to see the result every 10 interations beyond iteration 80
include_enc, exclude_enc, extras_enc = createEncodings(include, exclude, extras)
res_img, res_z=training_loop(Params, optimizer, show_crop=True)
```

Augmented cropped example



Generation



```
torch.cuda.empty_cache()
include=['A BLUE TREE IN THE FOREST', 'KIDS PLAYING IN MOON', 'FLOWERS DANCING']
exclude='watermark'
extras = ""
w1=1
w2=1
noise_factor=.20
total_iter=110
show_step=10 # set this to see the result every 10 interations beyond iteration 80
include_enc, exclude_enc, extras_enc = createEncodings(include, exclude, extras)
res_img, res_z=training_loop(Params, optimizer, show_crop=True)
```

Augmented cropped example



Generation



```
def interpolate(res_z_list, duration_list):
    gen_img_list=[]
    fps = 25

    for idx, (z, duration) in enumerate(zip(res_z_list, duration_list)):
        num_steps = int(duration*fps)
```